Adding Authentication, Authorization, and Access Controls with the Open Directory Framework

Directory Service Development in Snow Leopard

Session 606

These are confidential sessions—please refrain from streaming, blogging, or taking pictures.

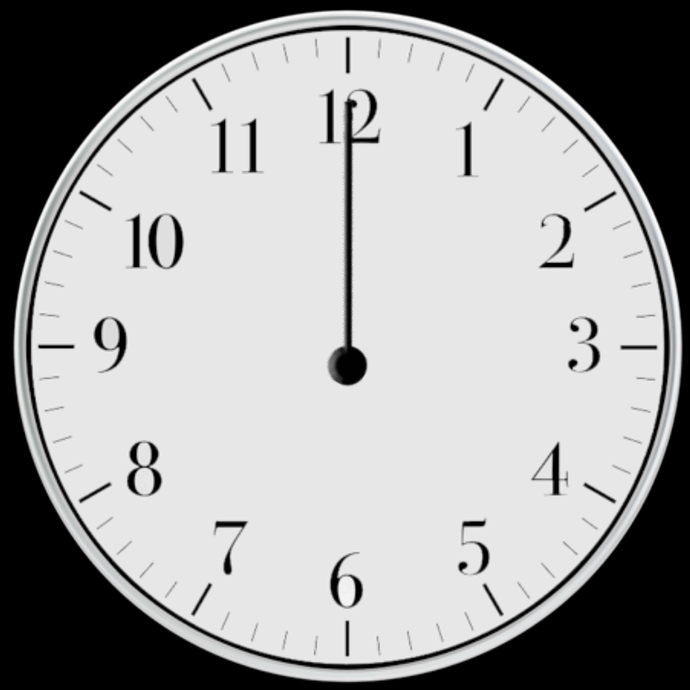
Adding Authentication, Authorization, and Access Controls with the Open Directory Framework

Directory Service Development in Snow Leopard

Nicole Jacque
Directory Server Engineering

Agenda

- Directory Services Overview
- Open Directory Framework
- Membership API
- Testing and Debugging
- Doing it right



Directory Services Overview

DirectoryService

- Translates data between the OS and datastores
  - Flat file, directory servers
- Interoperable Plugins
  - Open Directory
  - Active Directory
  - Unix flat files
  - NIS
  - eDirectory

OS X Directory Services

Who Uses It?

- Everybody!

Who Uses It?

- Everybody!
- Developers
  - Developers of services
  - Application Developers
  - In-house

Directory Service API

DirectoryService API

Deprecated

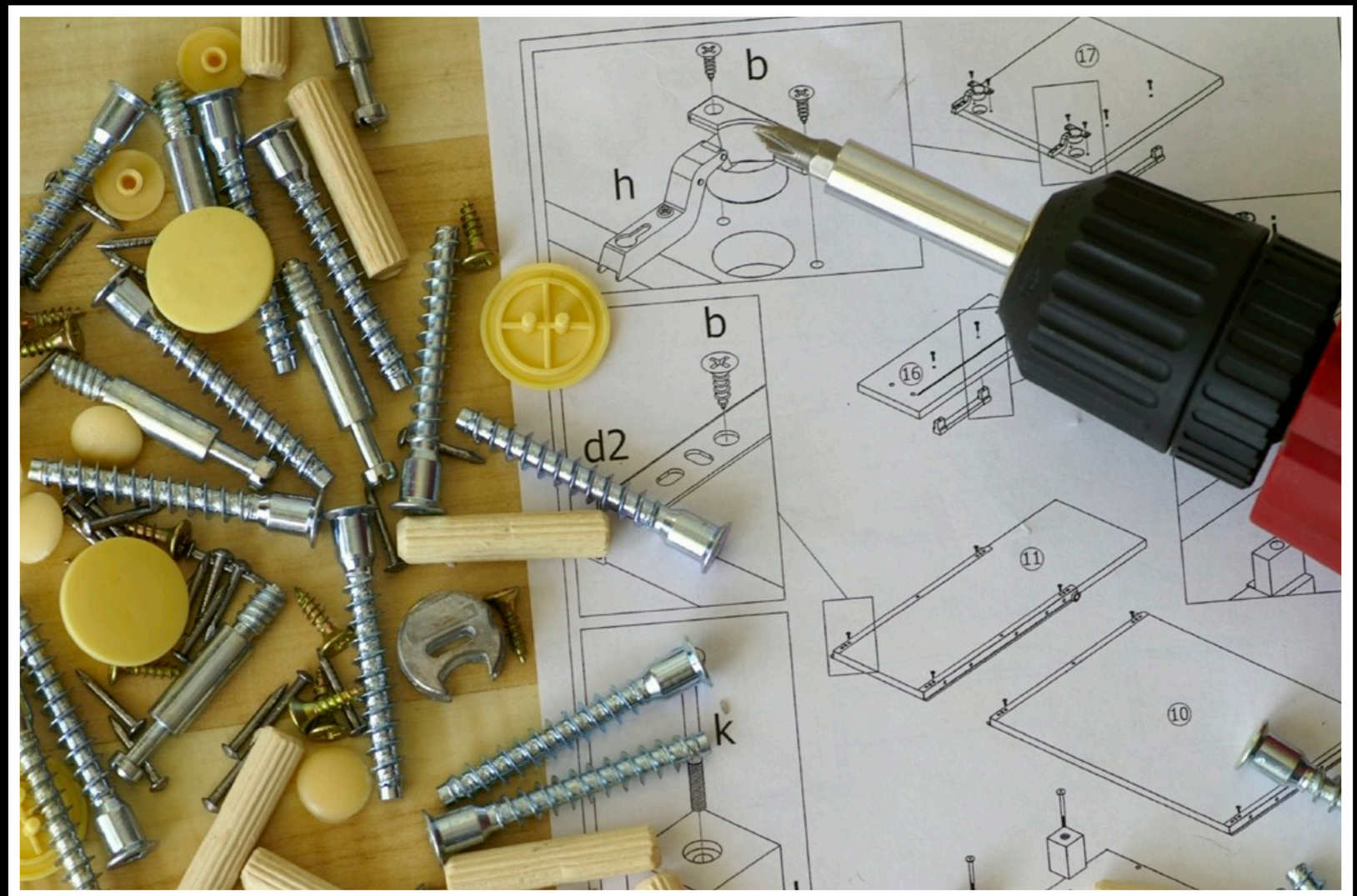
Open Directory Framework

Open Directory Framework

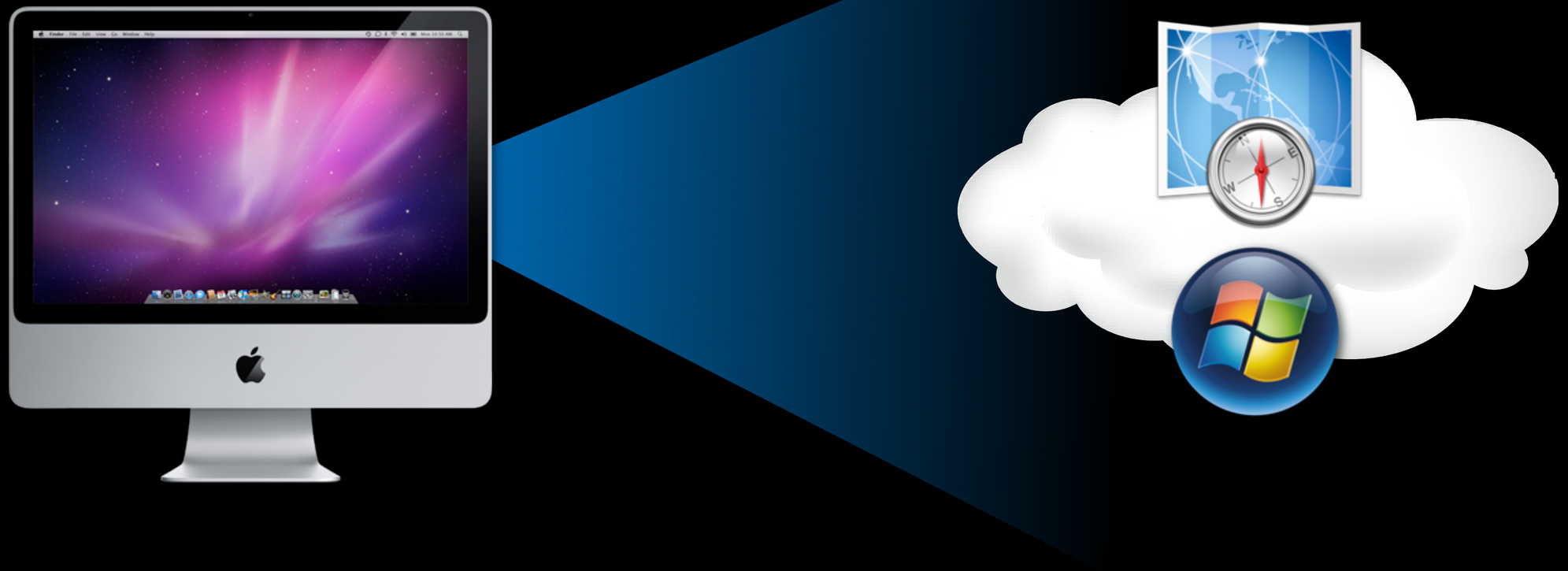
- High level API
- Comes in two flavors: Cocoa and CoreFoundation
- Read, write, authenticate to any DS node
  - Active Directory, Open Directory, Local, etc.
- Thread safe
- Simple

NSOpenDirectory Classes

- ODSession
- ODNode
- ODRecord
- ODQuery



ODSession



ODSession

session = [ODSession defaultSession];

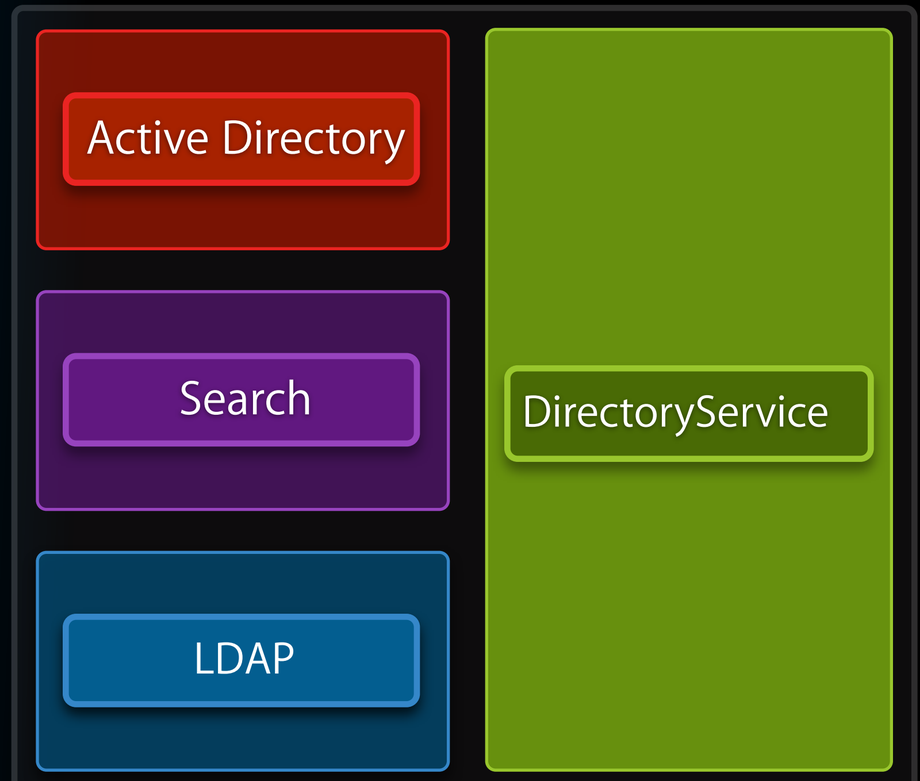
Getting Information About a Session

```
ODSession *session = [ODSession defaultSession];
```

```
// Get an array of nodes available in the session
NSArray *subNodes= [odSession subnodeNamesAndReturnError:outError];
```

```
// Get an Array of Nodes that are currently down
NSArray *unreachableNodes= [odSession unreachableSubnodeNamesAndReturnError:outError];
```

ODNode



- Datastore
  - Flat files or directory server
  - Can be a collection of datastores
    - Search node—nodes used for authentication
    - /Active Directory/All Domains—all configured AD nodes

ODNode

Specify Node by Name

/LDAPv3/server.com
/Active Directory/All Domains
/Search
/Local/Default

Active Directory

Search

LDAP

DirectoryService

```
node = [ODNode nodeWithSession:session name:@"/LDAPv3/myserver.com" error:&myError];
```

ODNode

Specify Node by Name

kODNodeTypeAuthentication
kODNodeTypeContacts
kODNodeTypeNetwork
kODNodeTypeLocalNodes
kODNodeTypeConfigure

Active Directory

Search

LDAP

DirectoryService

kODNodeTypeAuthentication = Search Node

ODNode

Specify Node by Name

kODNodeTypeAuthentication
kODNodeTypeContacts
kODNodeTypeNetwork
kODNodeTypeLocalNodes
kODNodeTypeConfigure

Active Directory

Search

LDAP

DirectoryService

```
node = [ODNode nodeWithSession:session type:kODNodeTypeAuthentication error:&myError];
```

Node Info

```
ODSession *session = [ODSession defaultSession];
```

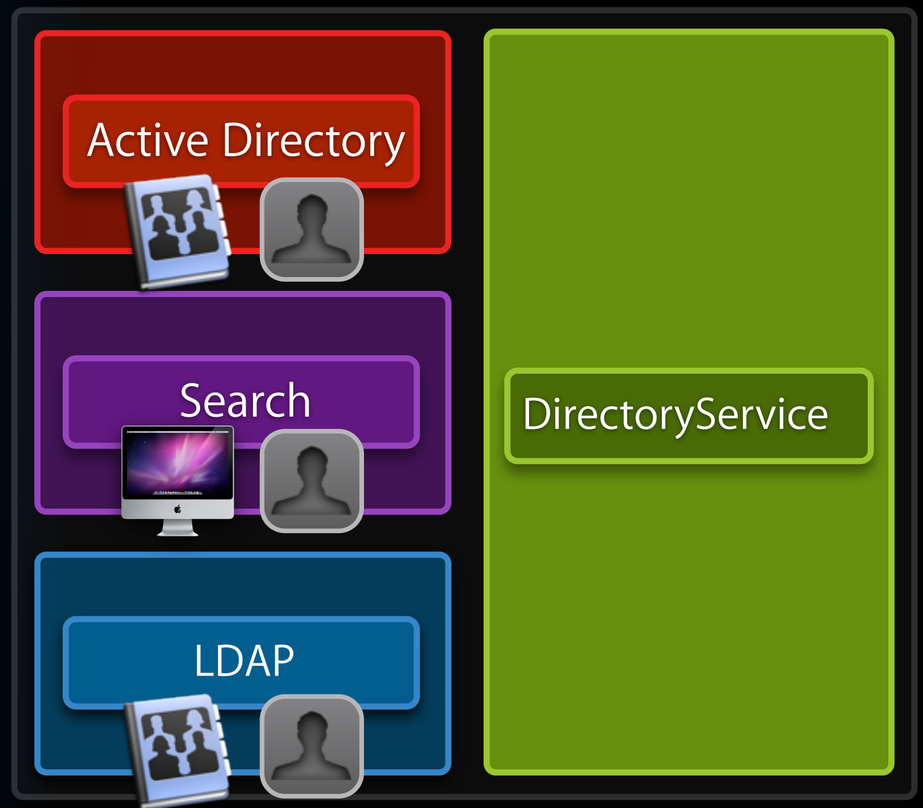
```
ODNode *node = [ODNode nodeWithSession:session type:kODNodeTypeAuthentication error:&authError];
```

```
// Get the name of the node
NSString *nodeName = [odNode nodeName];
```

```
// Get some info about the node
NSDictionary *nodeDetails = [node nodeDetailsForKeys:[NSArray arrayWithObject:@
kODAttributeTypeReadOnlyNode] error:nil];
```

```
// What record types does it support?
NSArray *recordTypes = [odNode supportedRecordTypesAndReturnError:outError];
```

ODRecord



A DirectoryService Record

RealName:	Nicole Jacque
NFSHomeDirectory:	/Users/nicole
Password:	********
Picture:	/Library/User Pictures/Sports/8ball.tif
PrimaryGroupID:	20
RecordName:	nicole najacque
RecordType:	dsRecTypeStandard:Users
UniqueID:	501

Workgroup Manager: Local

Server Admin | Accounts | Preferences | New User | Delete | Refresh | New Window | Search

Authenticated as diradmin to directory: /LDAPv3/127.0.0.1

Name contains

User Name	UID
Bertrand Serlet	1027
Clarus Dogcow	1032
Colonel Panic	1042
Cyber Dog	1039
Directory Administ...	1000
Issac Newton	1040
James Smith	1041
Jane Doe	1035
Joe User	1038
John Doe	1036
Johnny Appleseed	1033
Jonathan Ives	1034
Mary Nomdeplume	1037
Melody Sunshine	1044
Michael Souris	1043
Nicole Jacque	1030
Phil Schiller	1029
Steve Jobs	1026
Steve Wozniak	1031
Tim Cook	1028

1 of 20 users selected

Basic | Privileges | Advanced | Groups | Home | Mail | Print | Info | Windows

Name: Prefix | Steve | Mid | Jobs | Suffix

Address: 1 Infinite Loop

Cupertino

CA | 95014 | United States

Company: Apple Inc. | CEO | Department

Phone: Work 555 555-5555

Email: demosteve@apple.com

Chat: AIM demosteve

Homepage: http:/homepage.mac.com/steve/Menu8.html/stevejobs

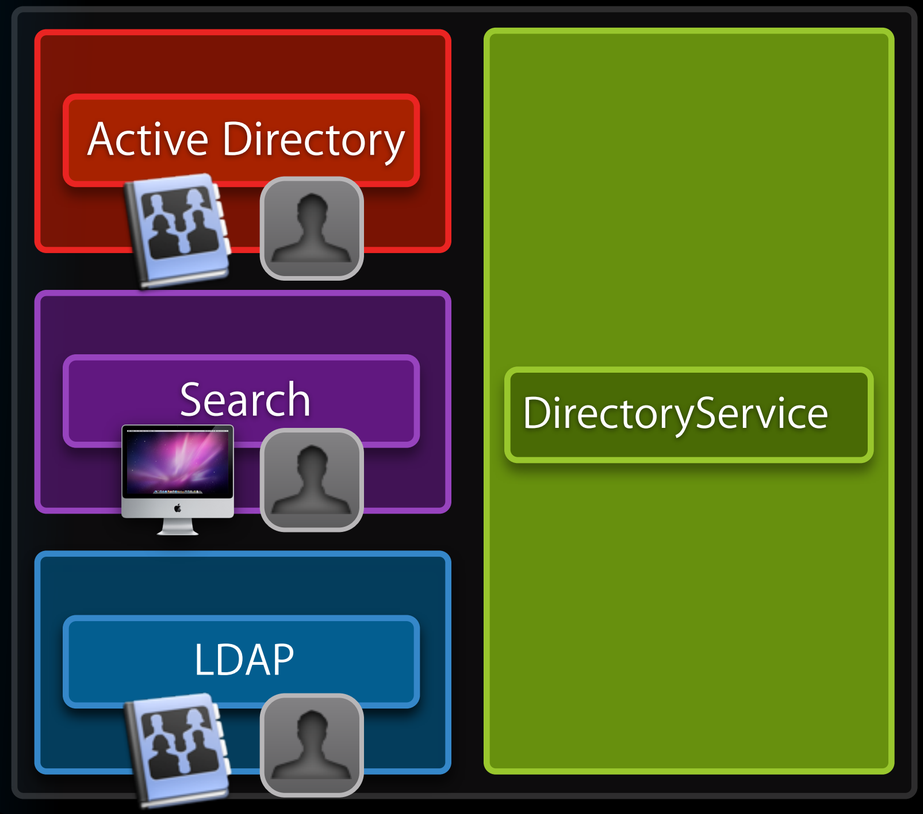
Weblog:

Presets: None

Revert | Save

ODRecord

kODRecordTypeUsers
kODRecordTypeGroups
kODRecordTypeComputers
kODRecordTypeConfiguration
kODRecordTypeMounts
kODRecordTypePeople



Getting a Record

When you already know the record name

```
ODSession *session = [ODSession defaultSession];
ODNode *node = [ODNode nodeWithSession:session type:kODNodeTypeAuthentication error:&outError];
ODRecord record = [node recordWithRecordType:kODRecordTypeUsers name:authUser attributes:[NSArray
arrayWithObject:kODAttributeTypeAllAttributes] error:&outError];

record = [node recordWithRecordType:kODRecordTypeUsers name:authUser attributes:[NSArray
arrayWithObject:kODAttributeTypeAllAttributes] error:&outError];
```

```
// read an attribute value from the record
NSArray *attribs = [record valuesForAttribute:kODAttributeTypeUserShell error:nil];
```

```
// or get a dictionary of all attribute values
// passing nil for the list of attributes returns all attributes
NSDictionary *attribDict = [record recordDetailsForAttributes:nil error:nil];
```

Useful Record Methods

Passwords

- verifyPassword
- changePassword

Useful Record Methods

Passwords

```
ODSession *session = [ODSession defaultSession];
ODNode *node = [ODNode nodeWithSession:session type:kODNodeTypeAuthentication
error:&authError];
ODRecord record = [node recordWithRecordType:kODRecordTypeUsers name:authUser attributes:
[NSArray arrayWithObject:kODAttributeTypeAllAttributes] error:&authError];

NSString *authPass = @"currentpass";
NSString *newPass = @"newpass";
```

```
// verify a user's password
[record verifyPassword:authPass error:nil]
```

```
// change the password
[record changePassword:authPass toPassword:newPass error:nil]
```

Editing Records

- Three steps
  - Set credentials on a node or record
    - Specify credentials for what account is making the change
  - Make changes to the record
  - Synchronize the record

Editing Records

```
// set the credentials
// ie. what user is editing the records?
BOOL authed = [someUser setNodeCredentials:adminUser password:password error:&error];
```

```
// make the change
[someUser addValue:plistData toAttribute:kODAttributeTypeComment error:&error];
```

```
// synchronize your change to the record
// for CFOpenDirectory, this would be ODRecordSynchronize()
[someUser synchronizeAndReturnError:&error];
```

Group Membership

- Check for membership in a group
- Edit group membership

Group Membership

```
ODSession *session = [ODSession defaultSession];
ODNode *node = [ODNode nodeWithSession:session type:kODNodeTypeAuthentication error:&outError];

ODRecord *userRecord = [node recordWithRecordType:kODRecordTypeUsers name:someUser attributes:[NSArray
arrayWithObject:kODAttributeTypeAllAttributes] error:&outError];
ODRecord *groupRecord = [node recordWithRecordType:kODRecordTypeGroups name:mygroup attributes:[NSArray
arrayWithObject:kODAttributeTypeAllAttributes] error:&outError];
```

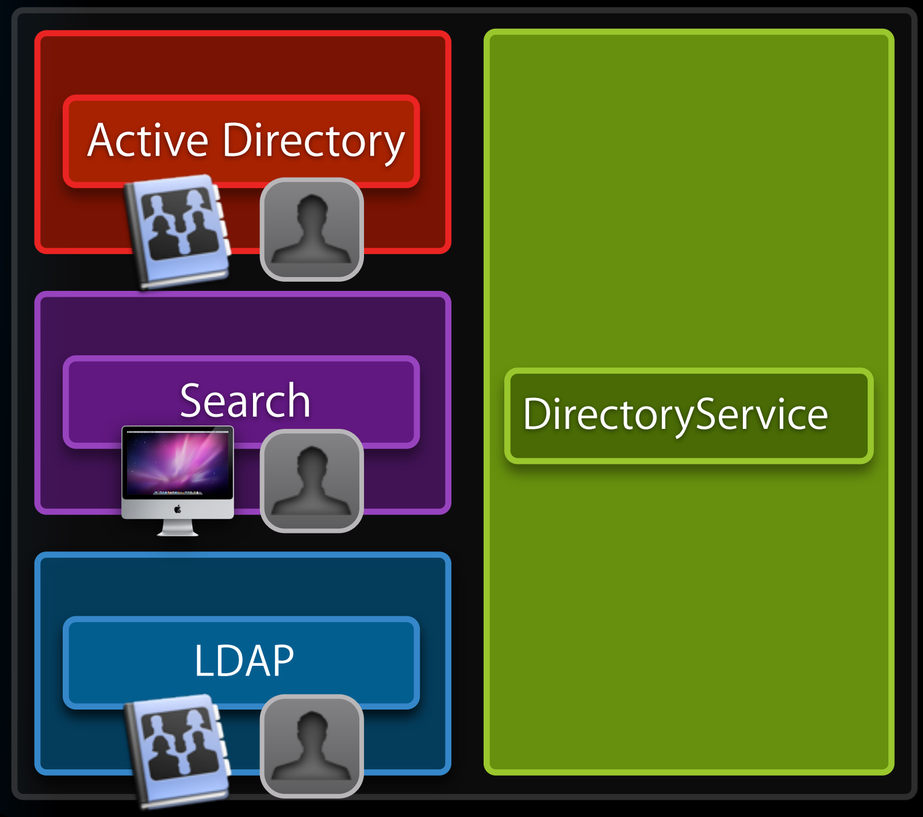
```
// add the user to the group
[groupRecord addMemberRecord:userRecord error:odErr];
[groupRecord synchronizeAndReturnError:&error];
```

```
//is the user in the group?
[groupRecord isMemberRecord:userRecord error:odErr];
```

```
// remove the user from the group
[groupRecord removeMemberRecord:userRecord error:odErr];
[groupRecord synchronizeAndReturnError:&error];
```

ODQuery

Find all Users with a phone number that begins with 555, and return their name, uid, street address, and zip



Running a Query

```
ODSession *session = [ODSession defaultSession];
ODNode *node = [ODNode nodeWithSession:session type:kODNodeTypeAuthentication error:&authError];
```

```
ODQuery *query = [ODQuery queryWithNode:node
                         forRecordTypes:kODRecordTypeUsers
                              attribute:kODAttributeTypeRecordName
                              matchType:kODMatchEqualTo
                            queryValues:shortNameString
                       returnAttributes:attributes
                         maximumResults:1
                                  error:&error];
NSArray *results = [query resultsAllowingPartial:NO error:&error];
```

Running Queries

With Blocking and Polling

```
NSArray *results = [query resultsAllowingPartial:NO error:&error];
```

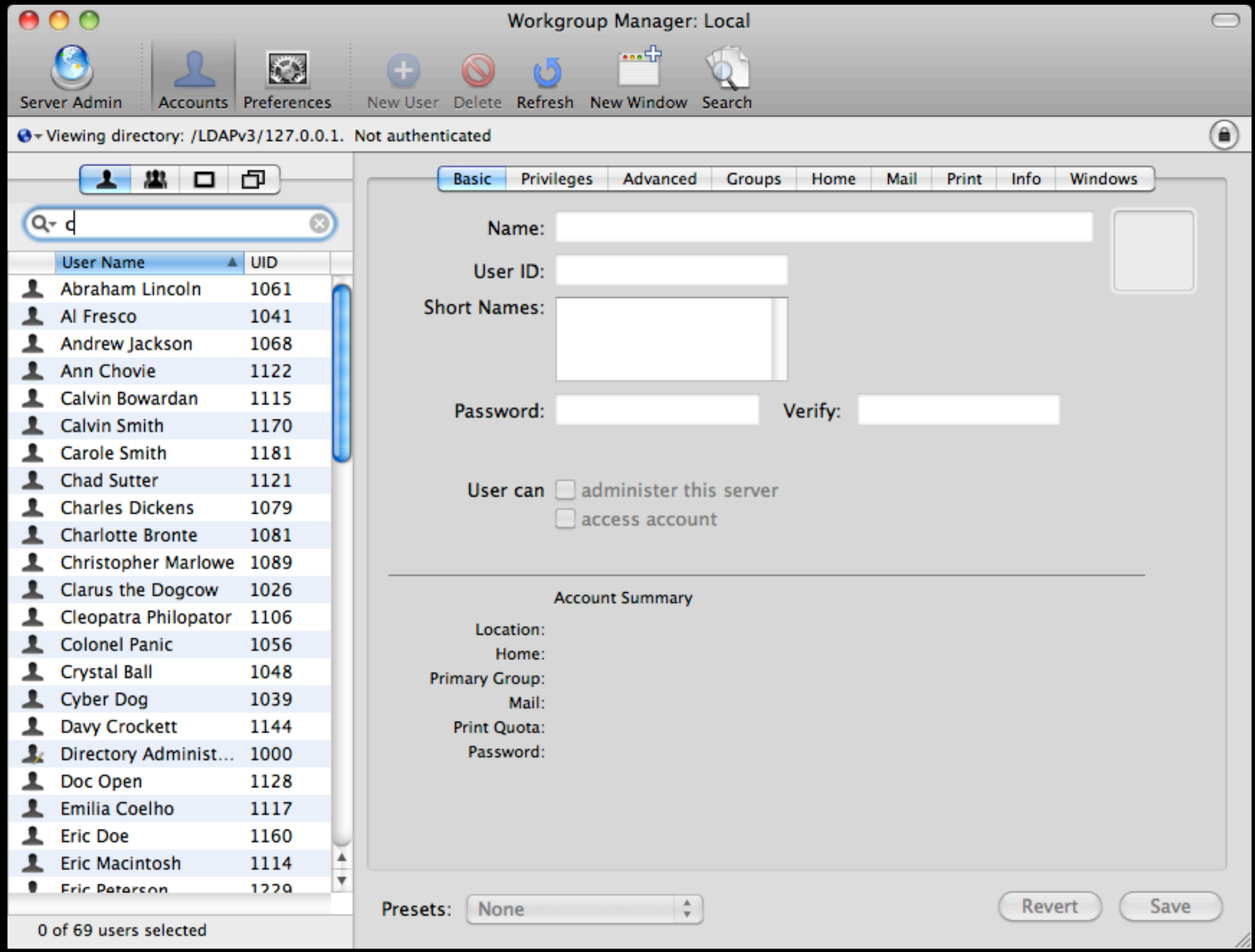
- NO = blocks until all results
  - Use asynchronous query if you don't want blocking
- YES = poll for results
  - Do not use resultsAllowingPartial:YES, use asynchronous query instead

[ODQuery synchronize]

- Dumps existing results, restarts query

Demo

Fun with records


Workgroup Manager: Local
Server Admin
Accounts
Preferences
New User
Delete
Refresh
New Window
Search
Viewing directory: /LDAPv3/127.0.0.1. Not authenticated
User Name
UID
Abraham Lincoln 1061
Al Fresco 1041
Andrew Jackson 1068
Ann Chovie 1122
Calvin Bowardan 1115
Calvin Smith 1170
Carole Smith 1181
Chad Sutter 1121
Charles Dickens 1079
Charlotte Bronte 1081
Christopher Marlowe 1089
Clarus the Dogcow 1026
Cleopatra Philopator 1106
Colonel Panic 1056
Crystal Ball 1048
Cyber Dog 1039
Davy Crockett 1144
Directory Administ... 1000
Doc Open 1128
Emilia Coelho 1117
Eric Doe 1160
Eric Macintosh 1114
0 of 69 users selected
Basic
Privileges
Advanced
Groups
Home
Mail
Print
Info
Windows
Name:
User ID:
Short Names:
Password:
Verify:
User can
administer this server
access account
Account Summary
Location:
Home:
Primary Group:
Mail:
Print Quota:
Password:
Presets:
None
Revert
Save

Queries

- Queries can be asynchronous
  - ODQueryDelegate protocol for receiving results

ODQueryDelegate

```
#import <Cocoa/Cocoa.h>
#import <OpenDirectory/OpenDirectory.h>

@interface ODController : NSObject <ODQueryDelegate> {

}

- (void)query:(ODQuery *)inQuery foundResults:(NSArray *)inResults error:(NSError *)inError;
@end
```

Asynchronous Query

```
ODSession *session = [ODSession defaultSession];
ODNode *node = [ODNode nodeWithSession:session type:kODNodeTypeAuthentication error:&authError];
```

```
ODQuery *myQuery = [[ODQuery queryWithNode:node
                forRecordTypes:kODRecordTypeUsers
                attribute:kODAttributeTypeFullName
                matchType:kODMatchInsensitiveContains
                queryValues:searchKey
                returnAttributes:kODAttributeTypeAllAttributes
                maximumResults:0
                error:&error] retain];
```

```
[myQuery setDelegate:self];
```

```
[myQuery scheduleInRunLoop:[NSRunLoop currentRunLoopforMode:NSDefaultRunLoopMode];
```

[ODQuery synchronize]

- For Asynchronous queries delegate is called with:
  - inResults == nil
  - [inError code] == kODErrorQuerySynchronize
  - [inError domain] == ODFrameworkErrorDomain
- The delegate should discard existing results and wait for new ones

Error Codes

Not the old DS -14XXX codes

kODErrorRecordReadOnlyNode	4000
kODErrorRecordPermissionError	4001

kODErrorRecordParameterError	4100
kODErrorRecordInvalidType	4101
kODErrorRecordAlreadyExists	4102
kODErrorRecordTypeDisabled	4103

kODErrorRecordAttributeUnknownType	4200
kODErrorRecordAttributeNotFound	4201
kODErrorRecordAttributeValueSchemaError	4202
kODErrorRecordAttributeValueNotFound	4203

Documentation

- Documentation on dev site
- https://developer.apple.com/snowleopard/library/documentation/Networking/Reference/OpenDirectoryFramework/Reference/reference.html

Gotchas

- NSOpenDirectory: Check the header doc to see which things are autoreleased
- Use the search node for reading data rather than specific nodes
- Use asynchronous (NS/CF RunLoop) queries to prevent blocking

Demo

Asynchronous Queries

Membership API

Membership API

- Group membership
- Identity conversions

Group Membership

- Group membership rules have changed over time
  - Once upon a time…
    - Groups were in flat files
    - You were limited in how many groups
    - Only users could be in groups

Times Have Changed…

- Things are more complex now
  - A group can have 1000's of users
  - A user can have 1000's of groups
  - You can nest groups
  - Computers can be in groups
  - Groups can span nodes

Employees

Engineering

Hardware

Software

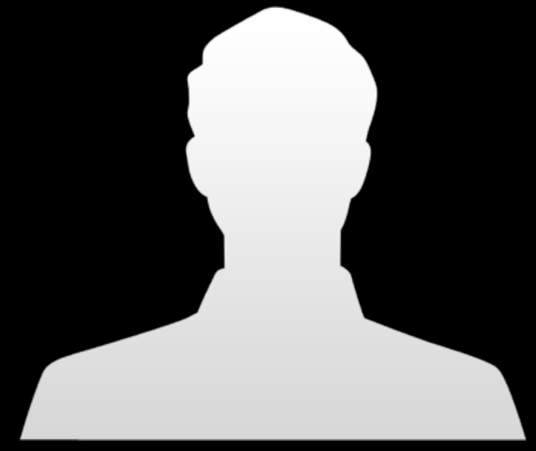
Marketing

Sales

Users Have Changed Too

- Users now have more “unique” identifiers

username	nicole
uid/gid	501
GUID/UUID	EFB9F97F-45B7-4043-B6BF-94631ABF440F
SID	S-1-5-21-2437042486-3009723068-1198334866-2002
Kerberos ID	nicole@APPLE.COM



Membership API to the Rescue

- /usr/include/membership.h
- Hides group structure
- Provides caching
- Translates between UID/GID, GUID, SID, other names

Checking Membership

```
error = mbr_check_membership(userGUID, groupGUID, &isMember);
```

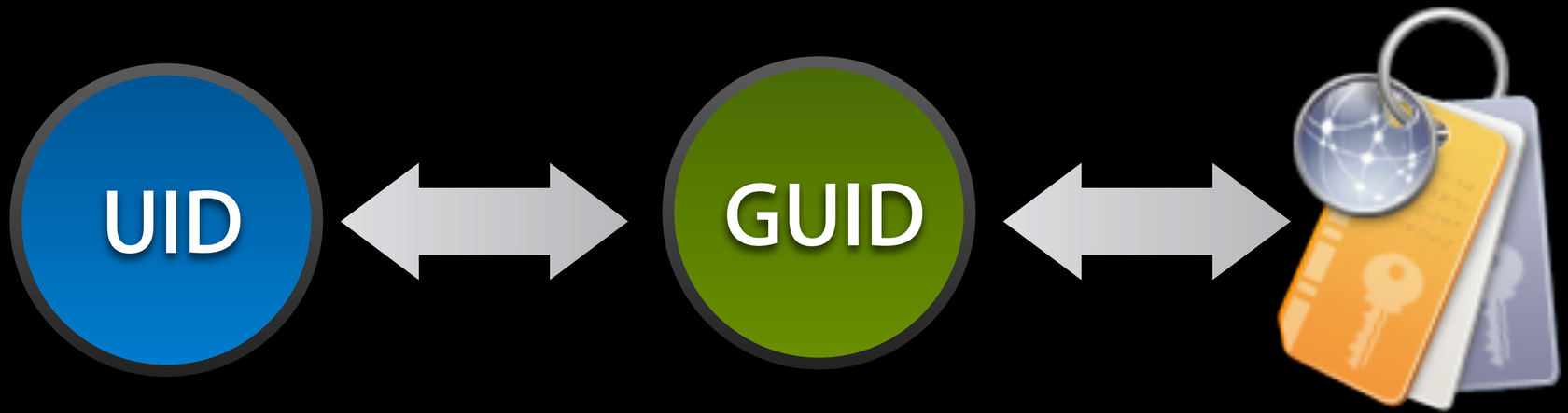
- isMember
  - 0 = is a member, or both GUIDs passed were equal
  - 1 = is not a member
  - error = 0 on success, EIO or ENOENT

Convert Between Identities

```
result = mbr_uid_to_uuid(uid, &userGUID);

result = mbr_uuid_to_sid(userUUID, &userSID);

mbrErr = mbr_identifier_to_uuid(ID_TYPE_KERBEROS, principal, strlen(principal), user_uuid);
```

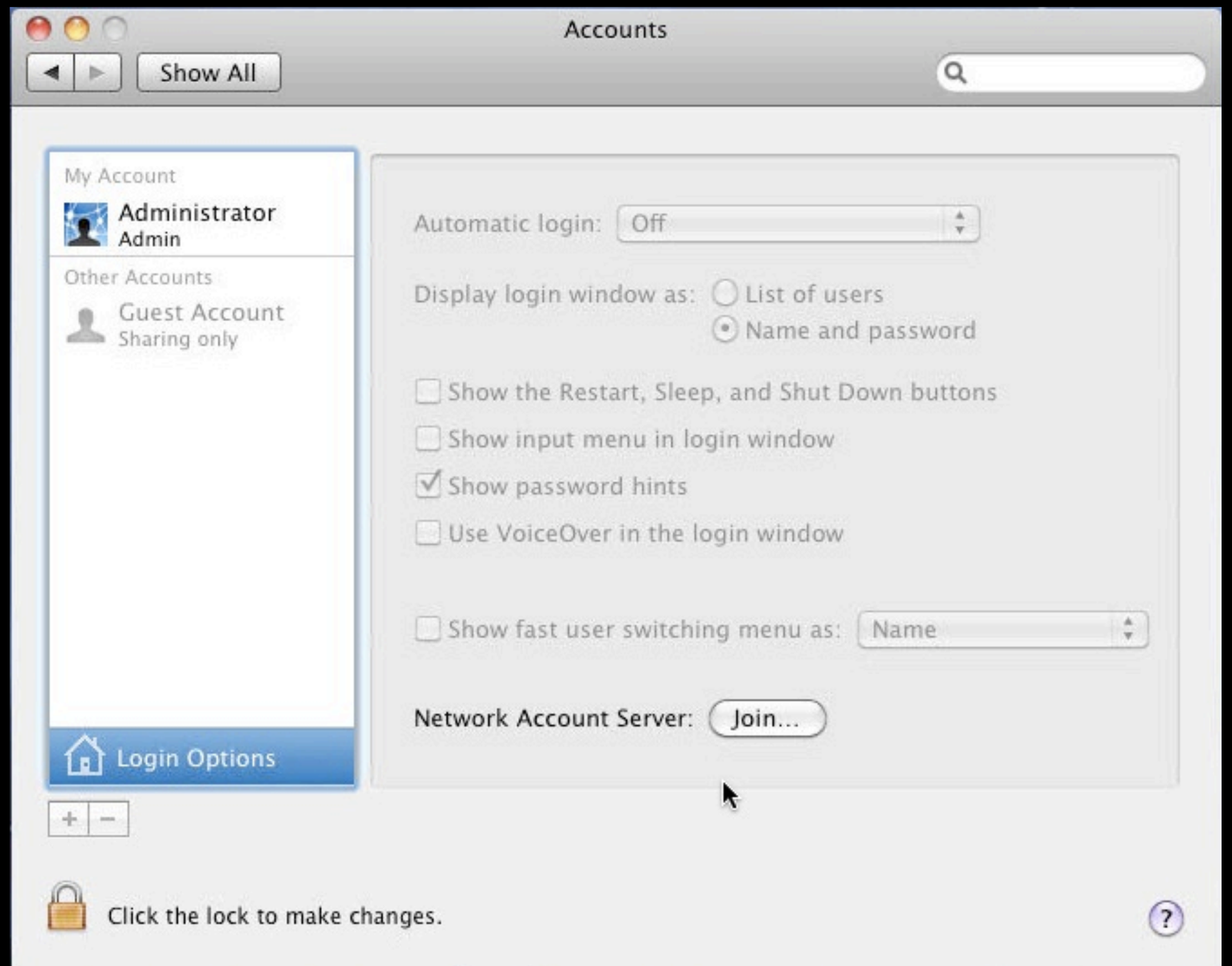


Why Use Membership API?

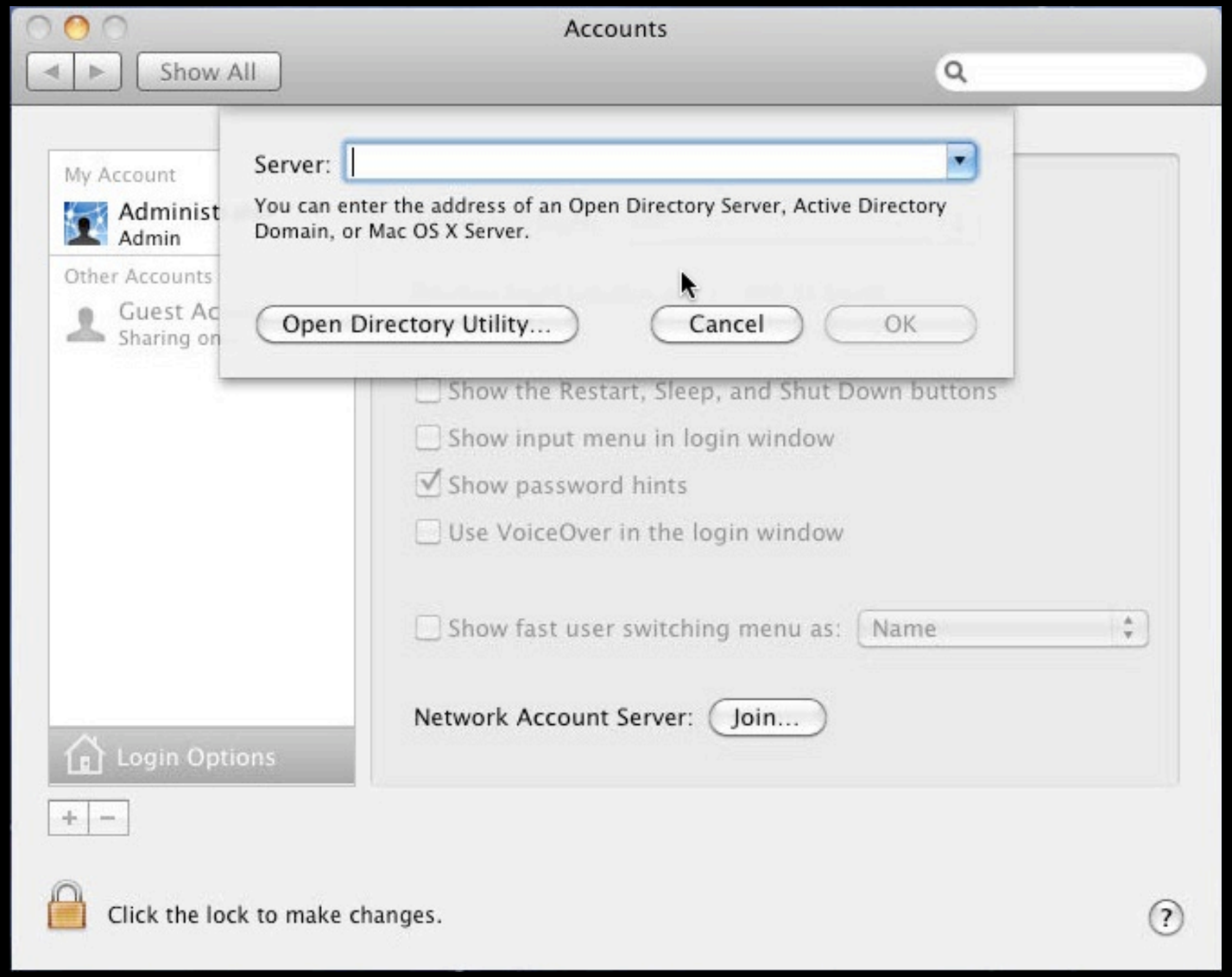
- Modern user and group functionality
- Performance
- It’s easy!

Testing and Debugging

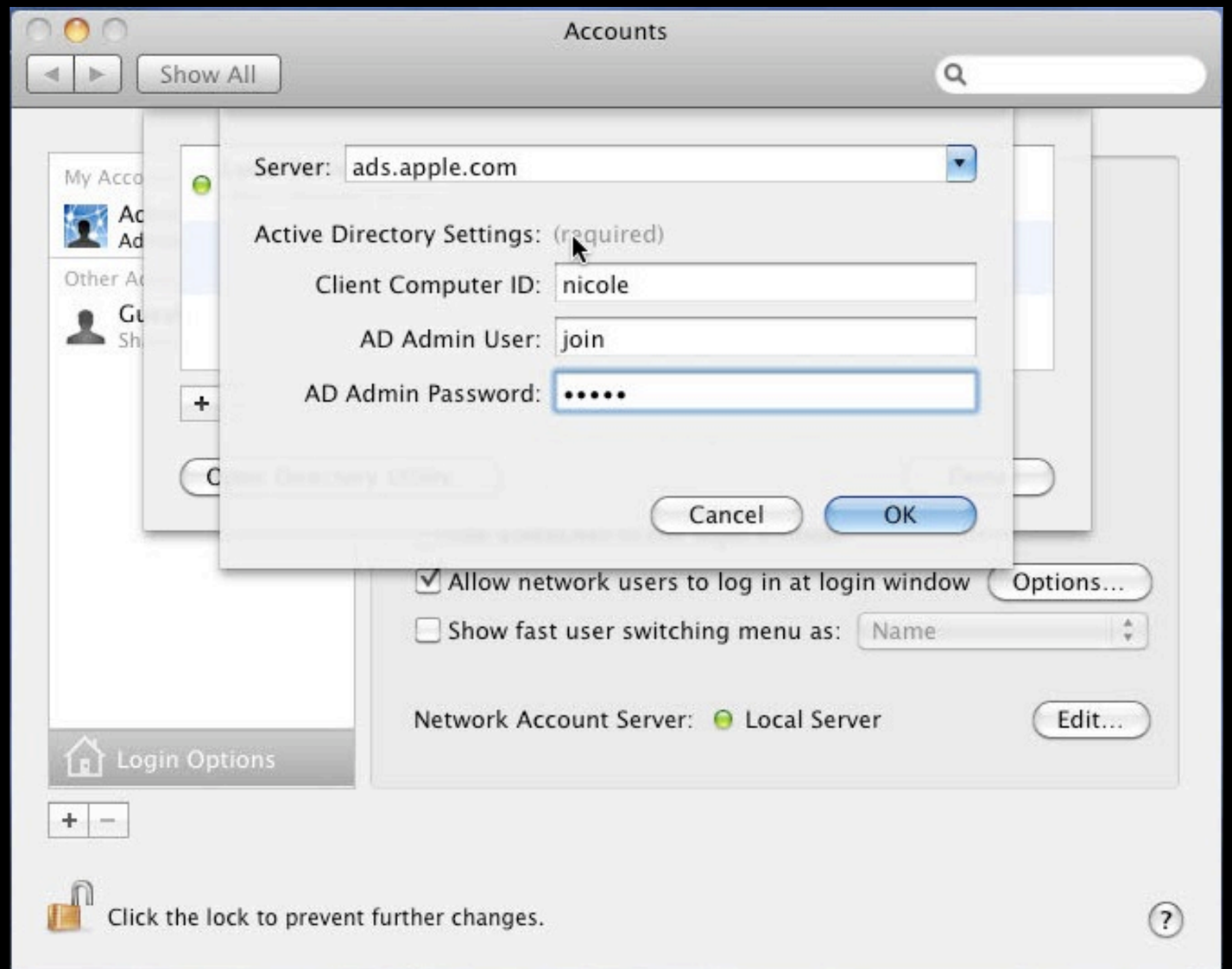
Configuring Directory Servers



Configuring Directory Servers



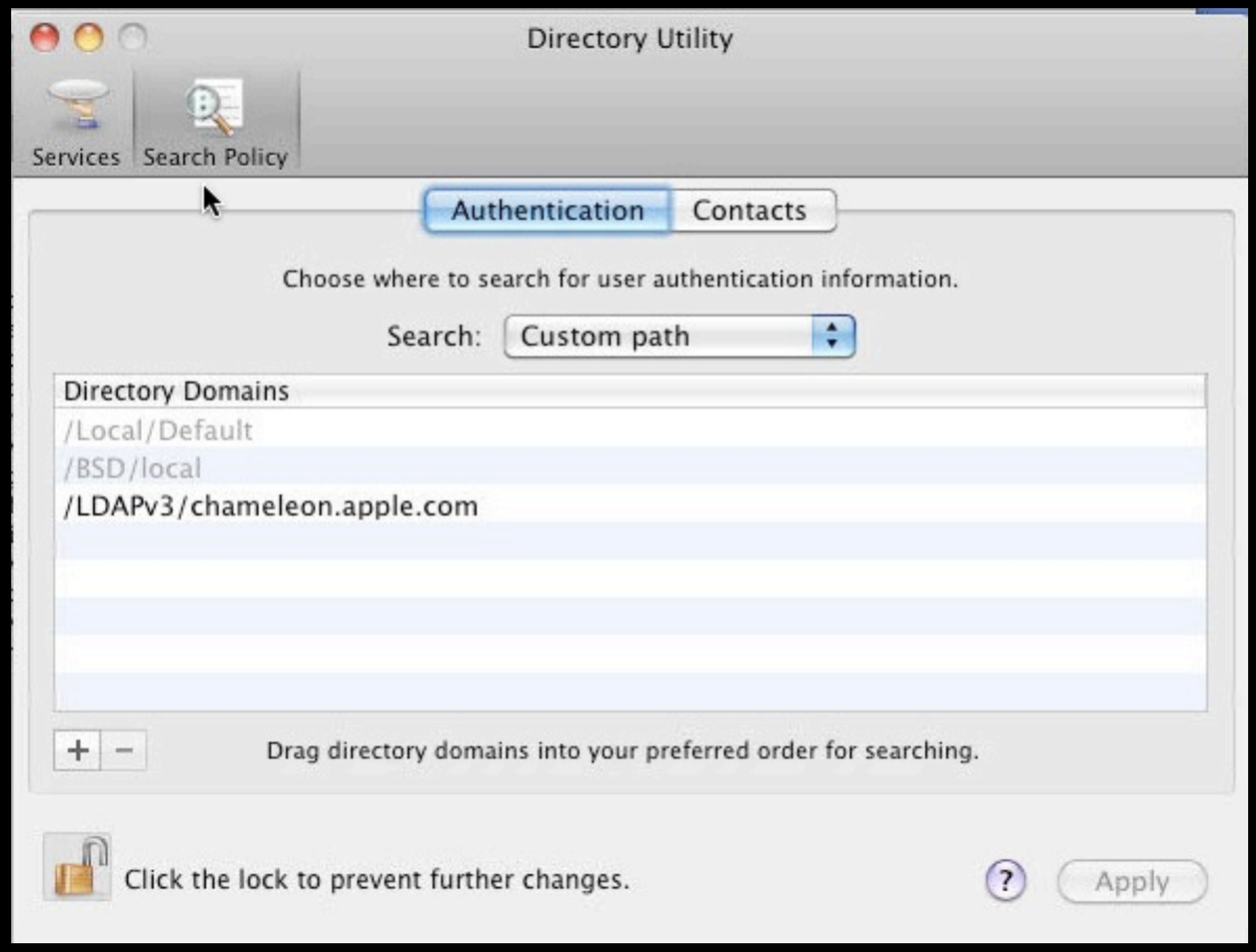
Configuring Directory Servers



Other Options

- dsconfigldap and dsconfigad
- Directory Utility still accessible for more complex binding requirements
  - Lives in /System/Library/CoreServices

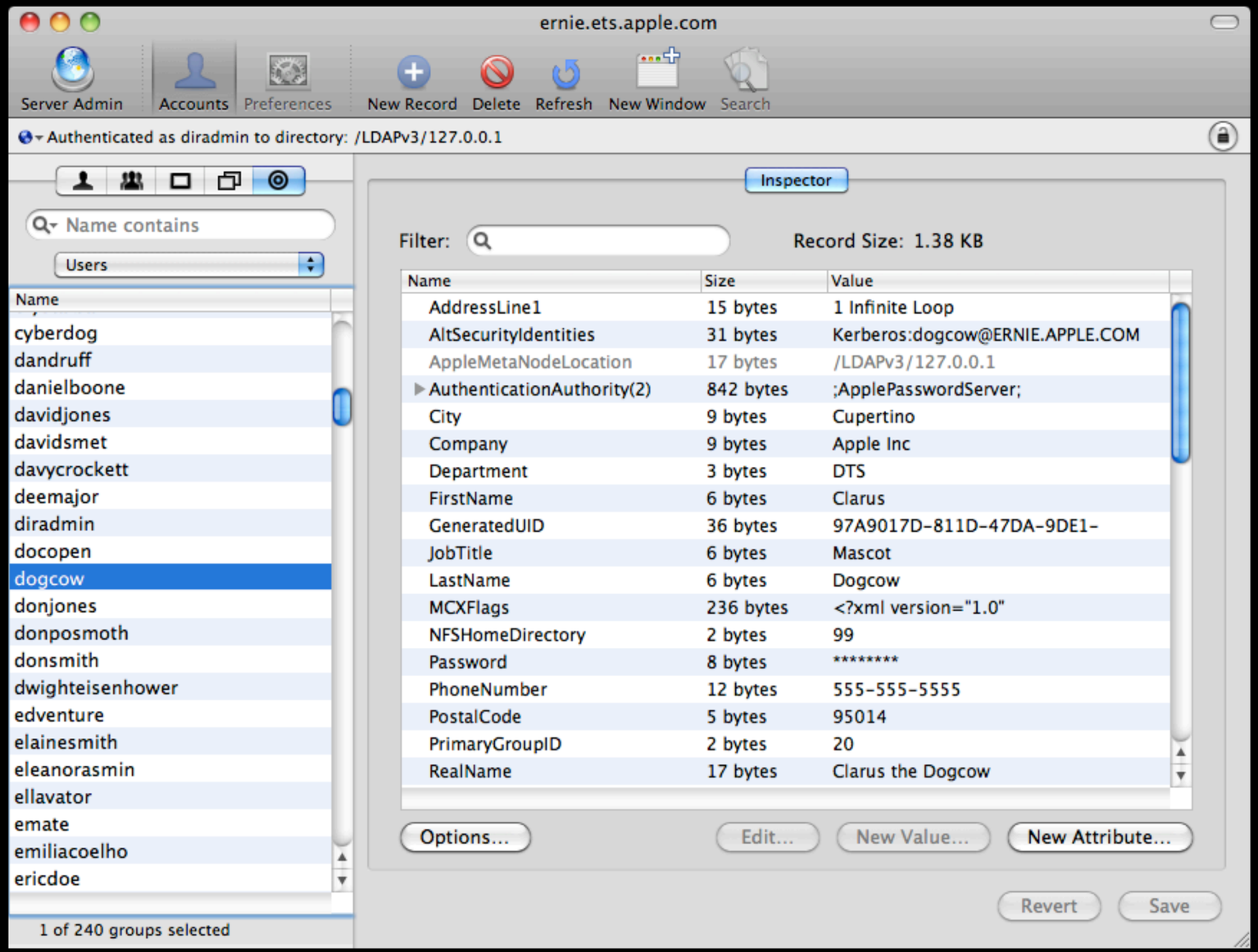
Verifying the Search Node



DS Tools

- Workgroup Manager
- id
- dscl
- dsmemberutil
- dseditgroup

Workgroup Manager



id

```
$ id admin
uid=501(admin) gid=20(staff) groups=20(staff),
103(com.apple.sharepoint.group.2),502(com.apple.local.ard_reports),
102(com.apple.sharepoint.group.1),101(com.apple.access_screensharing),
104(com.apple.sharepoint.group.3),504(com.apple.local.ard_manage),
204(_developer),100(_lpoperator),98(_lpadmin),81(_appserveradm),80(admin),
79(_appserverusr),61(localaccounts),12(everyone),
503(com.apple.local.ard_interact),500(com.apple.access_ssh),
501(com.apple.local.ard_admin)
```

dscl

- Search/look up records
- Test authentication
- Read/write data

dscl

```
$ dscl /Search -read Users/nicole
AppleMetaNodeLocation: /Local/Default
AuthenticationAuthority: ;ShadowHash;
;Kerberosv5;;nicole@LKDC:SHA1.9237127B69863BCC4E3D55C16DF6F4D48CE2EFCF;LKDC:SH
A1.9237127B69863BCC4E3D55C16DF6F4D48CE2EFCF;
AuthenticationHint:
GeneratedUID: EFB9F97F-45B7-4043-B6BF-94631ABF440F
NFSHomeDirectory: /Users/nicole
Password: ********
Picture:
 /Library/User Pictures/Sports/Baseball.tif
PrimaryGroupID: 20
RealName:
 Nicole Jacque
RecordName: nicole
RecordType: dsRecTypeStandard:Users
UniqueID: 501
```

dsmemberutil

- Flush the membership cache
- Check group membership

```
$dsmemberutil checkmembership -u 501 -G admin
```

- Do identity lookups

```
$dsmemberutil getuuid -U nicole
EFB9F97F-45B7-4043-B6BF-94631ABF440F
```

dseditgroup

- Manipulate groups from the command line

```
dseditgroup -o edit -n /LDAPv3/ldap.apple.com -u admin -p -a usertoadd -t user groupname
```

- Check membership

```
dseditgroup -o checkmember -m username groupname
```

Debug Logging

- Enable DS Debug logging
  - Manually turn logging on/off:

    ```
    sudo killall -USR1 DirectoryService
    ```

  - Start debugging automatically at startup:

    ```
    touch /Library/Preferences/DirectoryService/.DSLogDebugAtStart
    touch /Library/Preferences/DirectoryService/.DSLogDebugAtStartOnce
    ```

  - Change Logging Level (0-7):

    ```
    sudo defaults write /Library/Preferences/DirectoryService/
    DirectoryServiceDebug "Debug Logging Priority Level" -int 7
    ```

  - Logs file:

    ```
    /Library/Logs/DirectoryService/DirectoryService.debug.log
    ```

Per Process Debugging

DS Debug logs are complicated

```
2009-02-24 11:49:16 CST - T[0xB0289000] - Client: ipfw, PID: 202, API: libinfo, Server Used : libinfomig DAC : Procedure = getprotobynumber (13)
2009-02-24 11:49:16 CST - T[0xB0289000] - CCachePlugin::getprotobynumber - Cache hit for 17
2009-02-24 11:49:16 CST - T[0xB0289000] - Client: ipfw, PID: 202, API: libinfo, Server Used : libinfomig DAR : Procedure = getprotobynumber (13) : Result code = 0
2009-02-24 11:49:16 CST - T[0xB0103000] - Client: ipfw, PID: 202, API: libinfo, Server Used : libinfomig DAC : Procedure = getprotobynumber (13)
2009-02-24 11:49:16 CST - T[0xB0103000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), Search Used : DAC : 1 : Node Ref = 16777224 : Requested Attr Type = dsAttrTypeNative
2009-02-24 11:49:16 CST - T[0xB0103000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), Search Used : DAC : 2 : Node Ref = 16777224 : Requested Attrs = dsAttrTypeStandard:A
2009-02-24 11:49:16 CST - T[0xB0103000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), Local Used : DAC : 1 : Node Ref = 16777538 : Requested Attr Type = dsAttrTypeNative:
2009-02-24 11:49:16 CST - T[0xB0103000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), Local Used : DAC : 2 : Node Ref = 16777538 : Requested Attrs = dsAttrTypeStandard:Ap
2009-02-24 11:49:16 CST - T[0xB0103000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), Local Used : DAR : Node Ref = 16777538 : Number of Found Records = 0 : Continue Data
2009-02-24 11:49:16 CST - T[0xB0103000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), BSD Used : DAC : 1 : Node Ref = 16777539 : Requested Attr Type = dsAttrTypeNative:nu
2009-02-24 11:49:16 CST - T[0xB0103000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), BSD Used : DAC : 2 : Node Ref = 16777539 : Requested Attrs = dsAttrTypeStandard:Appl
2009-02-24 11:49:16 CST - T[0xB0103000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), BSD Used : DAR : Node Ref = 16777539 : Number of Found Records = 1 : Continue Data =
2009-02-24 11:49:16 CST - T[0xB0103000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), Search Used : DAR : Node Ref = 16777224 : Number of Found Records = 1 : Continue Dat
2009-02-24 11:49:16 CST - T[0xB0103000] - CCachePlugin::AddEntryToCacheWithKeys - Entry 0x0029C540 added for record 0x0029C4C0 with key p_proto:1 - TTL 3600
2009-02-24 11:49:16 CST - T[0xB0103000] - CCachePlugin::AddEntryToCacheWithKeys - Entry 0x0029C540 added key p_name:icmp
2009-02-24 11:49:16 CST - T[0xB0103000] - Client: ipfw, PID: 202, API: libinfo, Server Used : libinfomig DAR : Procedure = getprotobynumber (13) : Result code = 0
2009-02-24 11:49:16 CST - T[0xB0103000] - CCachePlugin::AddEntryToCacheWithKeys - Entry 0x0029C540 added key p_name:icmp
2009-02-24 11:49:16 CST - T[0xB0103000] - Client: ipfw, PID: 202, API: libinfo, Server Used : libinfomig DAR : Procedure = getprotobynumber (13) : Result code = 0
2009-02-24 11:49:16 CST - T[0xB0289000] - Client: ipfw, PID: 202, API: libinfo, Server Used : libinfomig DAC : Procedure = getprotobynumber (13)
2009-02-24 11:49:16 CST - T[0xB0289000] - CCachePlugin::getprotobynumber - Cache hit for 1
2009-02-24 11:49:16 CST - T[0xB0289000] - Client: ipfw, PID: 202, API: libinfo, Server Used : libinfomig DAR : Procedure = getprotobynumber (13) : Result code = 0
2009-02-24 11:49:16 CST - T[0xB0103000] - Client: ipfw, PID: 202, API: libinfo, Server Used : libinfomig DAC : Procedure = getprotobynumber (13)
2009-02-24 11:49:16 CST - T[0xB0103000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), Search Used : DAC : 1 : Node Ref = 16777224 : Requested Attr Type = dsAttrTypeNative
2009-02-24 11:49:16 CST - T[0xB0103000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), Search Used : DAC : 2 : Node Ref = 16777224 : Requested Attrs = dsAttrTypeStandard:A
2009-02-24 11:49:16 CST - T[0xB0103000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), Local Used : DAC : 1 : Node Ref = 16777538 : Requested Attr Type = dsAttrTypeNative:
2009-02-24 11:49:16 CST - T[0xB0103000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), Local Used : DAC : 2 : Node Ref = 16777538 : Requested Attrs = dsAttrTypeStandard:Ap
2009-02-24 11:49:16 CST - T[0xB0103000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), Local Used : DAR : Node Ref = 16777538 : Number of Found Records = 0 : Continue Data
2009-02-24 11:49:16 CST - T[0xB0103000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), BSD Used : DAC : 1 : Node Ref = 16777539 : Requested Attr Type = dsAttrTypeNative:nu
2009-02-24 11:49:16 CST - T[0xB0103000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), BSD Used : DAC : 2 : Node Ref = 16777539 : Requested Attrs = dsAttrTypeStandard:Appl
2009-02-24 11:49:16 CST - T[0xB0103000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), BSD Used : DAR : Node Ref = 16777539 : Number of Found Records = 1 : Continue Data =
2009-02-24 11:49:16 CST - T[0xB0103000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), Search Used : DAR : Node Ref = 16777224 : Number of Found Records = 1 : Continue Dat
2009-02-24 11:49:16 CST - T[0xB0103000] - CCachePlugin::AddEntryToCacheWithKeys - Entry 0x002D37F0 added for record 0x00263D50 with key p_proto:2 - TTL 3600
2009-02-24 11:49:16 CST - T[0xB0103000] - CCachePlugin::AddEntryToCacheWithKeys - Entry 0x002D37F0 added key p_name:igmp
2009-02-24 11:49:16 CST - T[0xB0103000] - Client: ipfw, PID: 202, API: libinfo, Server Used : libinfomig DAR : Procedure = getprotobynumber (13) : Result code = 0
2009-02-24 11:49:16 CST - T[0xB0289000] - Client: servermgrd, PID: 49, API: libinfo, Server Used : libinfomig DAC : Procedure = getnameinfo (28)
2009-02-24 11:49:16 CST - T[0xB0289000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), Search Used : DAC : 1 : Node Ref = 16777224 : Requested Attr Type = dsAttrTypeStanda
2009-02-24 11:49:16 CST - T[0xB0289000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), Search Used : DAC : 2 : Node Ref = 16777224 : Requested Attrs = dsAttrTypeStandard:A
2009-02-24 11:49:16 CST - T[0xB0289000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), Local Used : DAC : 1 : Node Ref = 16777538 : Requested Attr Type = dsAttrTypeStandar
2009-02-24 11:49:16 CST - T[0xB0289000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), Local Used : DAC : 2 : Node Ref = 16777538 : Requested Attrs = dsAttrTypeStandard:Ap
2009-02-24 11:49:16 CST - T[0xB0289000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), Local Used : DAR : Node Ref = 16777538 : Number of Found Records = 0 : Continue Data
2009-02-24 11:49:16 CST - T[0xB0289000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), BSD Used : DAC : 1 : Node Ref = 16777539 : Requested Attr Type = dsAttrTypeStandard:
2009-02-24 11:49:16 CST - T[0xB0289000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), BSD Used : DAC : 2 : Node Ref = 16777539 : Requested Attrs = dsAttrTypeStandard:Appl
2009-02-24 11:49:16 CST - T[0xB0289000] - Internal Dispatch, API: dsDoAttributeValueSearchWithData(), BSD Used : DAR : Node Ref = 16777539 : Number of Found Records = 1 : Continue Data =
```

Per process Debugging

- To get debug logging for just a single command:
  - As root, run /usr/sbin/DirectoryService -appledebug
  - In another terminal 'setenv DS_DEBUG_MODE 1'

Doing DS Right

Playing Nice with Everyone

- Directory Services can support some complex configs
  - Open Directory
  - Active Directory
  - Flat File
  - NIS

Supporting Active Directory

- AD does things a little differently
  - Naming conventions
  - Auth methods
  - Multidomain forests
  - Group membership
- AD plugin shields you from most of this

Do’s and Don’ts

- ☑ Use the Search node!
- ☑ Use common auth methods

  not all node types support all types of authentication
- ☒ Avoid relying on non-standard attributes
- ☒ Don’t try to pull all records
- ☒ Avoid triggering events that disrupt DirectoryService
- ☑ Check to see if nodes are available

Do's and Don'ts

- ☒ Don't make assumptions about number/name of Kerb/DNS realms
- ☑ Be careful with parsing

  MYDOMAIN\my group, MYDOMAIN\myuser
- ☑ Use Open Directory Framework or Membership API for group membership
- ☑ Use membership API for user identities

More Information

Mark Malone
Internet and Server Technologies Evangelist
mmalone@apple.com

Apple Training and Certification
http://training.apple.com

Related Sessions

Managing User Privileges and Operations with Authorization Services	North Beach Wednesday 3:00PM
What's New in Directory Services	Nob Hill Wednesday 9:00AM

Labs

Directory Services Lab	Informations Technologies Lab Thursday 2:00PM

Q&A

WWDC Session Survey

You can rate your sessions on the WWDC Attendee Site.

Survey forms are located on each session description page at:

developer.apple.com/wwdc/attendee

WWDC09